Paper

An Efficient Method for Private Network Management Using IP Address Translation

Member	Naonobu Okazaki	(Mitsubishi Electric Corporation)
Non-member	Yoshimasa Baba	(Mitsubishi Electric Corporation)
Non-member	Tetsuo Ideguchi	(Aichi Prefectural University)
Non-member	Ken-ichi Nakata	(Mitsubishi Electric Information Network Corporation)
Member	Mi Rang Park	(Mitsubishi Electric Corporation)
Non-member	Shoichiro Seno	(Mitsubishi Electric Corporation)

Constructions of networks with the private IP addresses, i.e., private networks, become very popular. In general, private addresses may overlap each other. When we manage network equipment in private networks by a remote network manager using network management protocols, each private network requires its own network manager because the manager can not be shared by overlapping private networks. This leads to problems of cost and inefficiency.

In this paper, we discuss an efficient method for network management of private networks using the concept of IP address translations. In this method, a new mechanism will be proposed, in which, the globally unique address mapping of SNMP data in address translation will be guaranteed. We will also discuss a trial implementation of the proposed method. By evaluation of network management system models, we will show that the proposed method is effective to reduce the cost and space of equipment in case of a large system including many private networks.

Keywords: Private Networks, Network Management, IP Address Translation, SNMP

1. Introduction

As networks have become increasingly large and complex, logically and geographically, network management has become more and more important. Network service providers manage network equipment or terminals on user networks from the remote manager to provide users value-added network services, to keep networks from any troubles or to maximize its efficiency. SNMP (Simple Network Management Protocol)[1] is a standard network management protocol on TCP/IP and widely used over the Internet.

For network service providers, as well as to provide stable services, to reduce the cost for network management is one of the most important matters when they manage numbers of customer's network. Because the cost for network management principally consists of the cost and space of management equipment, it is very important to reduce them.

One of the most compelling problems facing the IP Internet is IP address depletion. To solve this problem, constructions of networks with private IP addresses[2] (private networks) become very popular. In general, private addresses in one private network, which are used under a policy of each user, overlap addresses in other private networks.

When a network service provider manages network equipment in their customer's private networks, whose IP addresses may overlap each other, from the remote network manager using network management protocols, each private network requires its own network manager because the manager can not be shared by overlapping private networks. In case of a large system including n private networks, n network management equipments are needed. This leads to problems of cost and inefficiency.

In this paper, we discuss an efficient method for

network management of private networks using the concept of network address translation. In this method, a new mechanism will be proposed, in which, the globally unique address mapping of SNMP data in address translation will be guaranteed. We present a procedure to realize the mechanism using formal descriptions and give its application examples. It will be shown that we can use the global address space efficiently by the proposed method in cases that many managed private networks using large address space include few managed objects. We also discuss a trial implementation system construction of the proposed method. By estimation of network management system models, we will show that the proposed method is effective to reduce the cost and space of equipment in case of a large system including many private networks.

In the rest of this paper, section 2 briefly gives a concept of existing method of network address translation. In section 3, we introduce an extension of network address translation considering IP addresses in SNMP data, and an efficient management method for private networks is presented. Section 4 discusses an implementation and estimation of the proposed method. We will conclude the paper in section 5.

2. Network Address Translation

The network address translation is developed as a solution to IP address depletion. It is based on the concept of address reuse. This solution takes advantage of the fact that a very few hosts in a private network are communicating outside of the network at any given time. In this section, we describe basic function and some aspects of Network Address Translator (NAT)[3].

2.1 Basic function of NAT

NAT is a router function that is needed to be implemented only on the border router of a private network. Its basic operation is as follows. The addresses inside a private network can be reused by any other private network. For instance, a single class A address could be used by many private networks. At each exit point between a private network and the global network, NAT is installed. NAT has an address translation table including the one-to-one correspondence between some private addresses and global addresses.

For instance, in the example of Figure 1, both private networks A and B internally use class A address 10.0.0.0. Network A's NAT (NAT-a) is assigned the class C address 198.76.29.0, and network B's NAT (NAT-b) is assigned the class C address 198.76.28.0. As the class C addresses are globally unique no other NAT can use them.

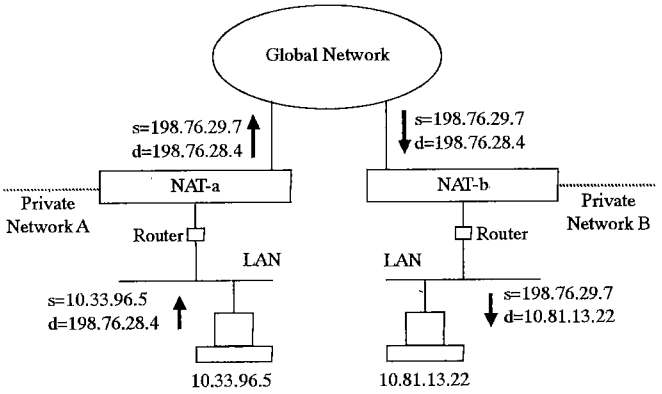


Fig. 1: Basic function of NAT[3].

When network A host 10.33.96.5 wishes to send a packet to network B host 10.81.13.22, it uses the globally unique address 198.76.28.4 as destination, and sends the packet to its primary router. The router has a static route for net 198.76.0.0 so the packet is forwarded to the global network. However, NAT-a translates the source address 10.33.96.5 of the IP header to the globally unique 198.76.29.7 before the packet is forwarded. Likewise, IP packets on the return path go through similar address translations. In this example, a private addresses 10.33.96.5 and 10.81.13.22 correspond to global addresses 198.76.29.7 and 198.76.28.4 in address translation tables in NAT-a and NAT-b, respectively.

2.2 Aspects of NAT

It is necessary to partition the IP address space into two parts - the reusable addresses used internal to private networks, and the globally unique addresses. We call the reusable address local addresses or private addresses, and the globally unique addresses global addresses.

In addition to modifying the IP address, NAT must modify the IP checksum and the TCP checksum. NAT must also look out for ICMP (Internet Control Message Protocol)[4] and FTP (File Transfer Protocol)[5] and modify the places where the IP address appears. There are undoubtedly other places, where modifications must be done.

3. A Management Method for Private Networks

3.1 Management of private networks using address translations

Figure 2 shows a network configuration for private network management. In this figure, the network manager manages two or more private networks. Each private network is connected to the network manager via the NAT. Each NAT translates private address space to unique address space without overlap with other global addresses.

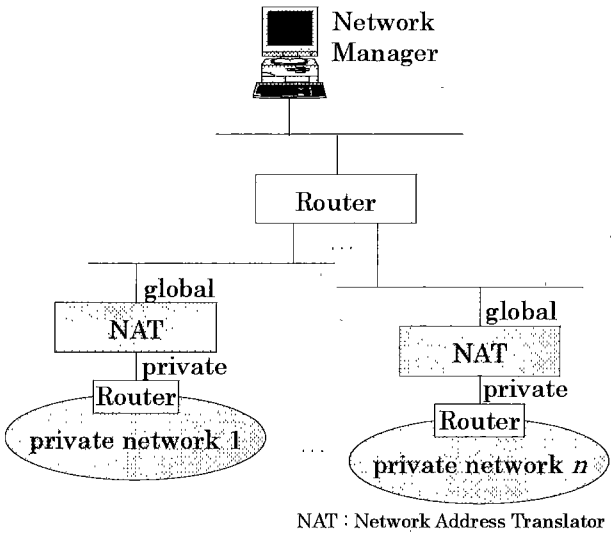


Fig. 2: A network architecture for private network management.

The SNMP[1] is used as a network management protocol.

In RFC 1631[3], there are no descriptions on modification of the IP address in SNMP data. An implementation of NAT without modification of the IP address in SNMP data causes some problems in private network management. For example, if the network manager detects an unknown network address in a router's MIB (Managed Information Base), the manager may conjecture inaccurate network configuration.

For example, consider the case in Figure 3, where the network manager "NM" gets MIB information on the interface "I/F 2" of the router "Router#2" using SNMP. In this case, NM sends an SNMP getrequest command requiring the information on the interface MIB of "I/F 2" to Router#2, and Router#2 responds with a getresponse command indicating the required information to NM. In the address translation table of NAT, address translation pairs are registered, in which "g_ip/g_mask <-> p_ip/p_mask" denotes that private address and its netmask "p_ip/p_mask" are translated to global address and its netmask "g_ip/g_mask", and vice versa.

Figure 4 shows the packet format of getrequest message transmitted from NM to Router#2. A source IP address, that is the IP address of NM, is included in the IP header field. In addition, Object ID fields may include the IP address of I/F 2 of Router#2. The source address in the IP header field will be translated by NAT from "10.0.30.1" to "192.168.20.1" in accordance with its address translation table. However, the IP address, if any, in Object ID fields will not be translated. In the case of getresponse message transmitted from Router#2 to NM, the IP address, if any, in Object ID field and the IP address or its netmask, if any, in the Value field will also not be translated.

Figure 5 shows a network configuration of Figure 3 conjectured inaccurately by the network manager. In this figure, two networks, 10.0.0.0/255.0.0.0 and 192.168.20.0/255.255.255.0, which should be concealed from the manager, are appeared as "dead" networks.

As mentioned above, IP addresses or netmasks in SNMP data field are not translated by NAT. This leads to the

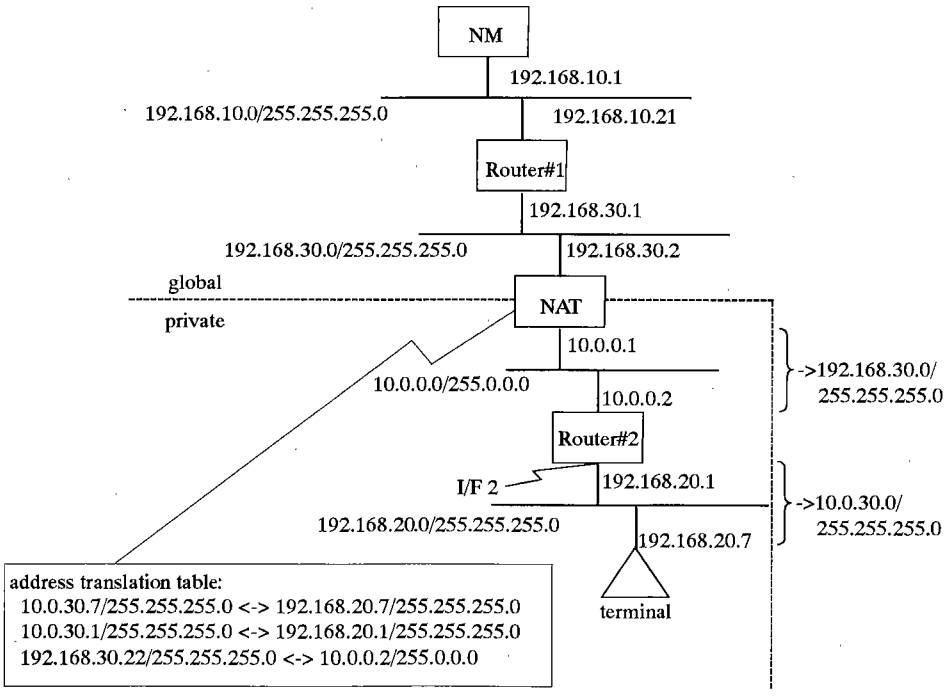


Fig. 3: An example of network management using NAT.

network manager unable to get correct information of managed equipment.

3.2 Address translation of SNMP data

To solve the above problem, the globally unique address mapping of SNMP data in address translation should be guaranteed. In addition, netmask associated with translated IP address should also be translated.

We can construct a private network using a large address space with private addresses. Indeed, there are many cases to use class A address for private networks, in which only a small number of addresses are needed. The global address space, however, is very restricted. Therefore, an address space of private networks should be reduced when it is translated to global address. This implies that netmask associated with translated IP address should be translated.

Consequently, the following is needed for existing NAT to provide transparency of SNMP data:

(1) To detect and translate target addresses appeared in SNMP data;
(2) To detect and translate netmask correspondent to the target addresses;
(3) Modification of length fields in SNMP data.

Additionally, a function to register and maintain a detected new address is needed.

3.3 Functions of address translation

We present a procedure to realize a mechanism of address translation in SNMP data using formal descriptions, and give its application examples.

3.3.1 Definitions

As preparations, we define the following notations.

[Def. 1] Address translation table Tr:

$Tr \subseteq G \times L$;

$G \subseteq Ga \times Gm$: global address/netmask pair;

$L \subseteq La \times Lm$: private address/netmask pair.

[Def. 2] Registered global address/netmask pair:

$Gtr = \{g \in G \mid \exists l \in L, (g,l) \in Tr\}$.

[Def. 3] Registered private address/netmask pair:

$Ltr = \{l \in L \mid \exists g \in G, (g,l) \in Tr\}$.

[Def. 4] Functions $global : L \to G, local : G \to L$:

$global(l) = g, local(g) = l$ if $(g,l) \in Tr$;

[Def. 5] Function $mask : Ga \to Gm$:

$mask(a) = m$ if $(a,m) \in Gtr \cup Ltr$.

[Def. 6] Instance Obj:

$Obj \subseteq Oid \times V$;

$Oid = \{id.ix \mid id \in Id, ix \in Ix\}$;

Id : Set of object identifier;

Ix : Set of index;

V : Set of value.

[Def. 7] Functions $id : Obj \to Oid, val : Obj \to V$:

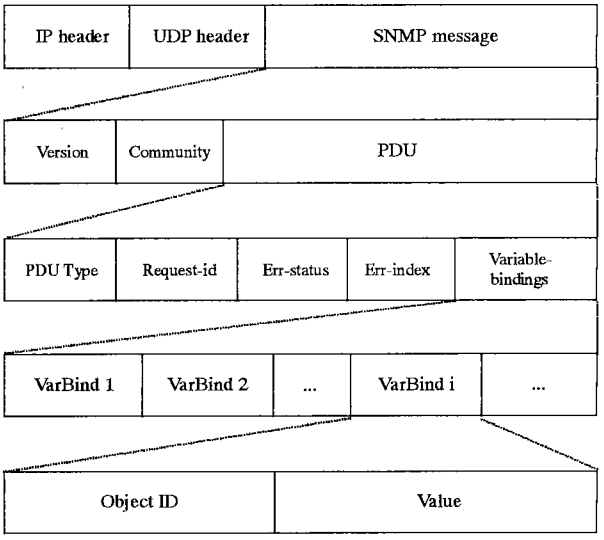


Fig. 4: SNMP message format.

$id(obj) = oid, val(obj) = v$

if $obj = (oid, v) \in Obj$.

[Def. 8] Functions $idfr : Oid \to Id, idx : Oid \to Ix$:

$idfr(oid) = id, idx(oid) = ix$

if $oid = id.ix \in Oid$.

[Def. 9] Set of object identifiers identifying objects consisting a value including an address registered in Tr:

$Ova = \{id \in Id \mid \exists obj \in Obj, \exists m \in Gm \times Lm,$
$idfr(id(obj)) = id$
$\wedge (val(obj), m) \in Gtr \times Ltr\}$.

[Def. 10] Set of object identifiers identifying objects consisting a value including a netmask registered in Tr:

$Ovm = \{id \in Id \mid \exists obj \in Obj, \exists a \in Ga \cup La,$
$idfr(id(obj)) = id$
$\wedge (a, val(obj)) \in Gtr \cup Ltr\}$.

[Def. 11] Set of object identifiers that consist of an index including an address registered in Tr:

$Oxa = \{id \in Id \mid \exists obj \in Obj, \exists m \in Gm \cup Lm,$
$idfr(id(obj)) = id$
$\wedge (idx(id(obj)), m) \in Gtr \cup Ltr\}$.

[Def. 12] Relation $R \subseteq Id \times Id$:

$R = \{(ida, idm) \mid \exists obja, objm \in Obj,$
$idx(id(obja)) = ida$
$\wedge idx(id(objm)) = idm$
$\wedge ((val(obja), val(objb)) \in Gtr \cup Ltr\}$.

[Def. 13] Function $refip : Ovm \to Ga \cup La$:

$refip(idm) = a$

if

$\exists obj \in Obj,$
$val(obj) = a$
$\wedge (idfr(id(obj)), idm) \in R$.

[Def. 14] Function $rc : Obj \to Obj$:

$obj' = rc(obj)$

if

obj' is an instance obj whose length fields are modified.

□

3.3.2 Procedure of address translation

The procedure of address translation, which is proposed in this paper, is shown here. The procedure consists of four steps: in step 1 and step 3, target IP addresses appeared in SNMP data within value field and object index, respectively, are translated; in step 2, netmask associated with the IP address is translated; step 4 is for modification of length fields in SNMP data.

[Address Translation Algorithm]

procedure
given : G, L, Tr, R;
input : obj;
output : obj;
start;
/* *step 1* */
if $idfr(id(obj)) \in Ova\{$
  if $(val(obj), m) \in Gtr$
    $val(obj) \leftarrow local(val(obj))$
  else if $(val(obj), m) \in Ltr$
    $val(obj) \leftarrow global(val(obj))\}$
/* *step 2* */
else if $i = idfr(id(obj)) \in Ovm\{$
  if $(refip(i), val(obj)) \in Gtr$
    $val(obj) \leftarrow mask(local(refip(i)))$
  else if $(refip(i), val(obj)) \in Ltr$
    $val(obj) \leftarrow mask(global(refip(i)))$
/* *step 3* */
if $idfr(id(obj)) \in Oxa\{$
  if $(idx(obj), m) \in Gtr$
    $idx(obj) \leftarrow local(idx(obj))$
  else if $(idx(obj), m) \in Ltr$
    $idx(obj) \leftarrow global(idx(obj))\}$
/* *step 4* */
$obj \leftarrow rc(obj)$;
end;

□

3.3.3 Examples

Application examples of the above procedure will be shown here. We consider the cases of two kinds of getresponse messages from Router#2 to NM in Figure 3.

In these cases, G, L, Tr, Ova, Ovm, Oxa and R are given as follows :

G={10.0.30.1, 10.0.30.7, 192.168.30.22},
L={192.168.20.1, 192.168.20.7, 10.0.0.2},
Tr={((10.0.30.1,255.255.255.0),(192.168.20.1,255.255.255.0)), ... },
Ova={ {1.3.6.1.2.1.4.20.1.1}, ... },
Ovm={ {1.3.6.1.2.1.4.20.1.3}, ... },

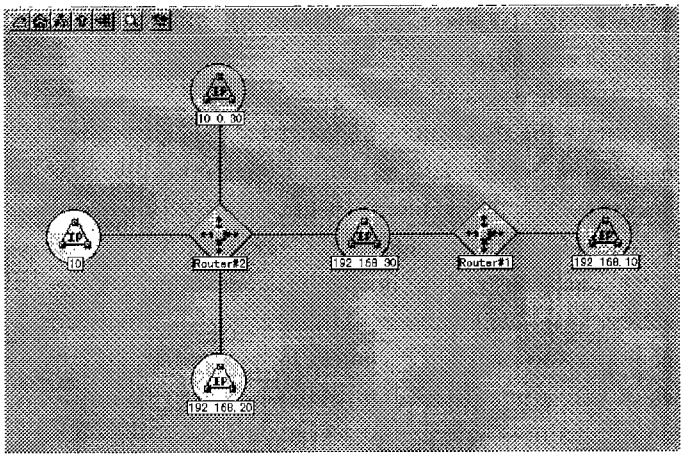


Fig. 5: Network configuration of Fig. 3 conjectured inaccurately by the network manager.

Oxa={ {1.3.6.1.2.1.4.20.1.1}, {1.3.6.1.2.1.4.20.1.2}, {1.3.6.1.2.1.4.20.1.3}, ... },
R={ ({1.3.6.1.2.1.4.20.1.3},{1.3.6.1.2.1.4.20.1.1}), ... }.

The first example shows the case where IP addresses in object ID and the value field of SNMP data are translated.

Example 1:
Consider an input :
obj = 0x
06 0F
2B 06 01 02 01 04 14 01 01 81 40 81 28 14 01
(name 1.3.6.1.2.1.4.20.1.1.192.168.20.1)
40 04
C0 A8 14 01
(value 192.168.20.1).
Here,
$idfr(id(obj)) = \{1.3.6.1.21.4.20.11\} \in Ova$ and
$(val(obj), m) = (192.168.20.1, 255.255.255.0) \in Ltr$.
Then,
$val(obj) \leftarrow global(val(obj)) = \{10.0.30.1\}$.
In addition, from
$idfr(id(obj)) = \{1.3.6.1.2.1.4.20.1.1\} \in Oxa$ and
$(idx(obj), m) = (192.168.20.1, 255.255.255.0) \in Ltr$,
$idx(obj) \leftarrow global(idx(obj)) = \{10.0.30.1\}$.

Consequently, output obj will be :
obj=0x
06 0D
2B 06 01 02 01 04 14 01 01 A0 00 1E 01
(name 1.3.6.1.2.1.4.20.1.1.10.0.30.1)
40 04
A0 00 1E 01
(value 10.0.30.1).

□

The next example shows the case where netmask in the value field is translated.

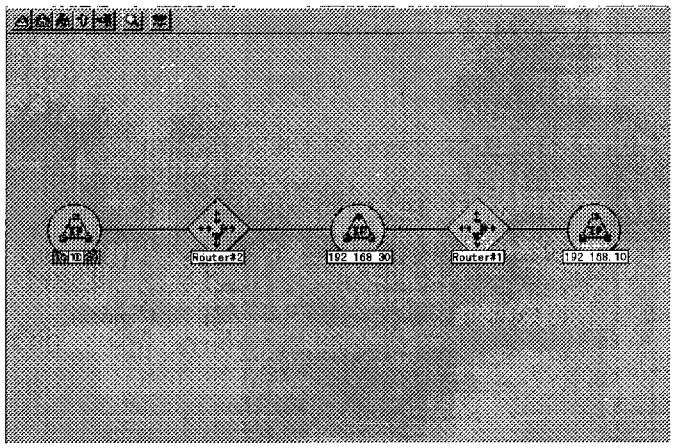


Fig. 6: Network configuration of Fig. 3 which the network manager conjectures correctly using the proposed method.

Example 2:
Consider an input :
obj=0x
06 0F
2B 06 01 02 01 04 14 01 03 81 40 81 28 14 01
(name 1.3.6.1.2.1.4.20.1.3.192.168.20.1)
40 04
FF FF FF 00
(value 255.255.255.0).
From
$i = idfr(id(obj)) = \{1.3.6.1.2.1.4.20.1.3\} \in Ovm$ and
$(refip(i), val(obj))$
$= (192.168.20.1, 255.255.255.0) \in Ltr$,
$val(obj) \leftarrow mask(global(refip(i)))$.
$= mask(global(192.168.20.1))$
$= mask(10.0.30.1)$
$= \{255.255.255.0\}$.

In addition, from
$idfr(id(obj)) = \{1.3.6.1.2.1.4.20.1.3\} \in Oxa$ and
$(idx(obj), m)$
$= (192.168.20.1, 255.255.255.0) \in Ltr$,
$idx(obj) \leftarrow global(idx(obj)) = \{10.0.30.1\}$.
Consequently, output obj will be :
obj=0x
06 0D
2B 06 01 02 01 04 14 01 03 A0 00 1E 01
(name 1.3.6.1.2.1.4.20.1.3.10.0.30.1)
40 04
FF FF FF 00
(value 255.255.255.0).

□

As a result, NM will be able to conjecture a "correct" network configuration shown in Figure 6 using the proposed procedure.

Note that, in this example, a class A *private* network

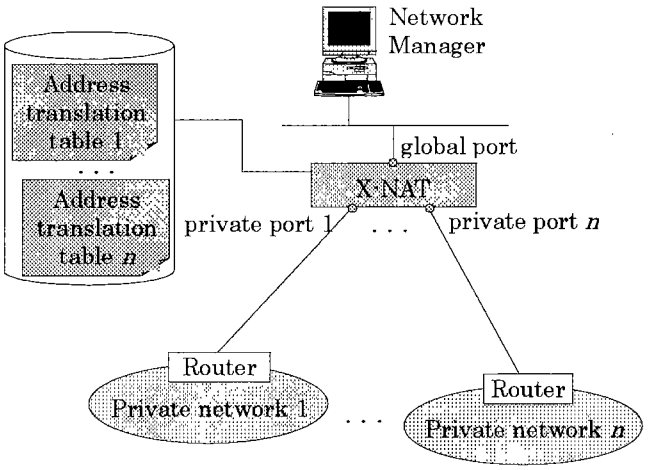


Fig. 7: Implementation system construction.

10.0.0.0/255.0.0.0 is translated to class C *global* network 192.168.30.0/255.255.255.0. This means that we can reduce the space of global addresses. By using the proposed method, we can use the global address space efficiently in cases where many managed private networks using a large address space include few managed objects.

4. Implementation and Estimation

In this section, we describe a trial implementation and an example system of an efficient private network management using the proposed method. It is shown that we can reduce the cost and space for network management of private networks.

4.1 Implementation system

A trial implementation system construction of the proposed method is shown in Figure 7. In this system, we introduce an extended NAT (X-NAT), which has a *global* port connecting to the network manager and *private* ports for private networks. Note that conventional NAT has only one port for a private network and translate source/destination IP address in IP header. In addition to the functions of NAT, X-NAT has the function of IP address translation in SNMP data. Because X-NAT has multiple private interfaces, it can treat a number of private networks. This leads to a very simple construction of network management for private networks.

Figure 8 shows the software construction of X-NAT. In this figure, SNMP-AT and IP-AT are the modules to translate IP addresses in SNMP DATA and IP header, respectively. In general, network addresses on the private ports may conflict each other, so X-NAT has ARP tables corresponding to those ports if the ports are connected to Ethernet. SNMP-AT and IP-AT refer to the common Address Translation Table. SNMP-AT is a new module implementing the functions described in section 3.3, and

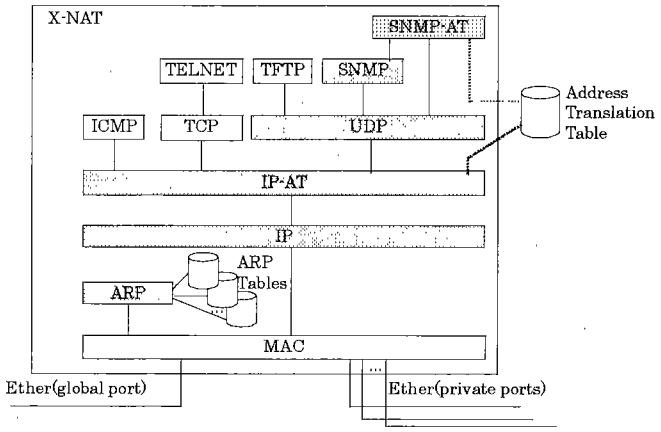


Fig. 8: Software construction of X-NAT.

SNMP, UDP, IP and ARP modules need to be modified from existing modules.

4.2 Estimation

We estimate the proposed management system model in comparison with an existing model in viewpoints of cost and space of equipment, because the cost for network management principally consists of them. Here, we compare three types of network management system models, i.e., existing model, NAT$^+$ model and X-NAT model based on the proposed method, which are shown in Figure 9. In NAT$^+$ model, NAT$^+$ has a partial function of X-NAT. That is, it can translate IP addresses in SNMP data, but has only one private interface. In this figure, it is shown that only one network manager is needed for managing private networks by using NAT$^+$ or X-NAT. Furthermore, X-NAT can be shared by more than one private networks.

Table 1 shows the result of estimation. In this table, m, n and c denote the number of private n etworks, the number of private interfaces on X-NAT and the cost of development of NAT$^+$ or X-NAT, respectively. The costs for Manager or NAT/X-NAT include the cost of its hardware and software. We assume that the manager is a workstation type one and NAT and X-NAT are based on PC without display monitors. From a typical architecture of PCs, number of PCI interfaces included in each PC is as many as 6, so it is reasonable to put n as 5. In this case, the cost and space of X-NAT model is estimated as 5% and 4% of the existing model under the large m, respectively.

It is shown that the proposed method is effective to reduce the cost and space of equipment in case of a large system including many private networks.

5. Conclusion

In this paper, we discussed an efficient method for network management of private networks using the concept of address translation where the network manager

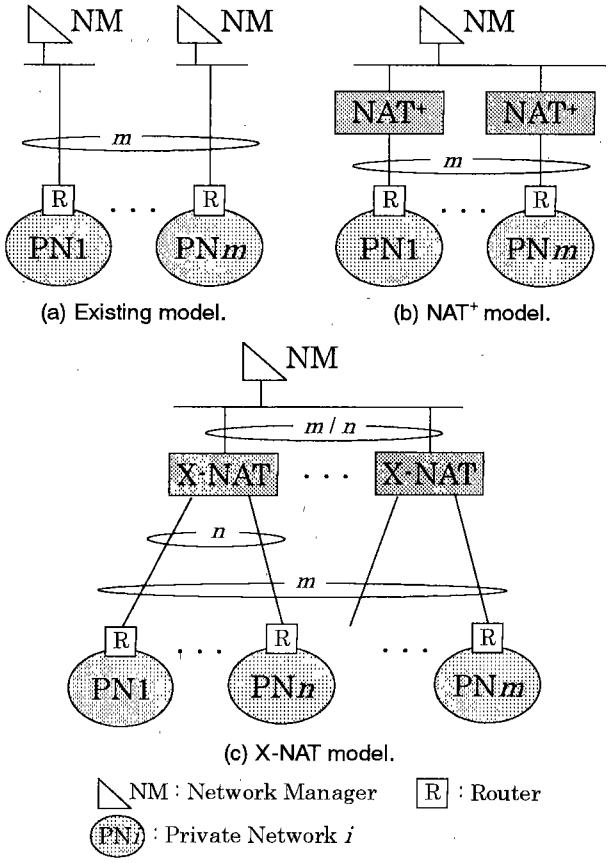


Fig. 9: Network management system models.

	Existing model	NAT$^+$ model	X-NAT model
Num. Manager(s)	m	1	1
Num. NAT/X-NAT	-	m	m / n
Cost per Manager (Million Yen)	2.0		
Cost per NAT/X-NAT (Million Yen)	-	$0.5+c*n / m$	
Total Cost (Million Yen)	$2m$	$2+(0.5+c*n / m)m$	$2+(0.5+c*n / m)m / n$
Space volume per Manager(m^3)	1.0		
Space volume per NAT/X-NAT(m^3)	-	0.2	
Total space volume(m^3)	m	$1+0.2m$	$1+0.2m / n$

Table 1: Estimation results.

deals with two or more private networks. In this method, a new mechanism is proposed, in which, the globally unique address mapping of SNMP data in address translation will be guaranteed. We presented the procedure to realize the mechanism using formal descriptions and gave its application examples. It was shown that we can use global address spaces efficiently by the proposed method in cases where many managed

private networks using a large address space included few managed objects.

We also discussed a trial implementation system construction of the proposed method and its software construction. From the results of estimation, it is shown that the proposed method is effective to reduce the cost and space of equipment in case of a large system including many private networks.

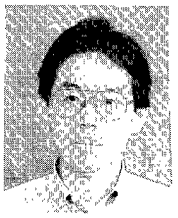
As future studies, we will consider applying it to a large system over the Internet such as an extranet. We will also consider mechanisms for remote control of address translation.

(Manuscript received May 30, 2000;
Revised June 4, 2001)

References

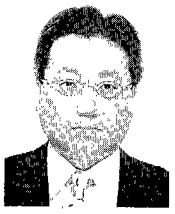
(1) J. Case, M. Fedor, M. Schoffstall and J. Davin, "A Simple Network Management Protocol (SNMP)", RFC1157, 1990.
(2) Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot and E. Lear, "Address Allocation for Private Internets", RFC1918, 1996.
(3) K. Egevang and P. Francis, The IP Network Address Translator (NAT)", RFC1631, 1994.
(4) J. Postel, "Internet Control Message Protocol", RFC792, 1981.
(5) Postel, J. and J. Reynolds, "File Transfer Protocol (FTP)", RFC959, 1985.
(6) N. Okazaki, Y. Baba and T. Ideguchi, "An Efficient Management Method of Private Networks", Proc. Multimedia, Distributed, Cooperative and Mobile Symposium, IPSJ, pp.101-107, 1998.
(7) N. Okazaki, Y. Baba, M. R. Park, S. Seno, K. Nakata and T. Ideguchi, "A Method for Private Network Management Using Address Translations", Proc. 1999 Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT' 99), pp.256-260, 1999.

Naonobu Okazaki (Member) received the B. S., M .S. and D. Eng. degree from Tohoku University, Sendai, Japan, in 1986, 1988 and 1992, respectively. He has been a senior researcher with Mitsubishi Electric Corporation, Information Technology R&D Center, Kamakura, Japan. His research interests include network management and formal description technique of communication systems. Dr. Okazaki is a member of the IPSJ, IEICE and IEEE.



Yoshimasa Baba (Non-member) received the B. S. and M. S. degree from Keio University in 1984 and 1986, respectively. In 1986 he joined Mitsubishi Electric Corporation and has been a manager of LAN systems team, Information Technology R&D Center. His research interests include network security and the development of network equipment. Mr. Baba is a member of the IPSJ.

Tetsuo Ideguchi (Non-member) graduated from University of Electro-Communications, Japan, in 1972 and received the D. Eng. degree from Tohoku University, Sendai, Japan, in 1993. He has been a Professor with Aichi Prefectural University, Japan, since 1998. His research interests include Network Architecture, Network Management and Mobile Communication. Dr. Ideguchi is a member of the IPSJ, IEICE and IEEE.



Ken-ichi Nakata (Non-member) received the B. S. degree from Osaka Electro-Communication University, Japan, in 1986. In 1989 he joined Mitsubishi Electric Information Network Corporation and has been a manager of network services section. His research interests include network management and network system architecture.

Mi Rang Park (Member) received the B. S. and M .S. degree from Han Yang University, Seoul, Korea, in 1983 and 1985, respectively, and D. Eng. degree from Tohoku University, Sendai, Japan, in 1993. She has been a senior researcher with Mitsubishi Electric Corporation, Information Technology R&D Center, Kamakura, Japan. Her research interests include network security and formal description technique of communication systems. Dr. Park is a member of the IPSJ.

Shoichiro Seno (Non-member) received the B. S. and M. S. degree from Tokyo Institute of Technology, Japan, in 1981 and 1983, respectively. In 1983 he joined Mitsubishi Electric Corporation. His research interests include network security and the development of network equipment. Mr. Seno is a member of IPSJ and IEICE.