网络工作组
RFC：1631
种类：信息

K. Egevang
Cray 通迅
P. Francis
NTT
1994 年 5 月

IP 网络地址转换（NAT）

本文的现状

本文提供供给因特网协会。本文没有规定定任何类型的因特网标准。发布本文没有限制。

摘要

当前 IP 因特网面临的两大严重问题是 IP 地址枯竭和路由的膨胀。长期和短期解决方案已经开发出来。短期解决方案是 CIDR（无类域间路由）。长期解决方案建议采用具有更大地址空间的新的互连协议。

CIDR 不适合继续支持 IP 因特网直到长期解决方案取代。本文建议采用另一种短期解决方案，地址重用，来作为 CIDR 的补充甚至取代它。地址重用解决方案要在根域的边界处安装网络地址转换器（NAT）。每个 NAT 盒有一个由成对的本地 IP 地址和全球唯一地址组成的表。在根域内部的 IP 地址并不是全球唯一的。它们在其它域中可重用，这样就可解决地址枯竭的问题。全球唯一的 IP 地址通过现在的 CIDR 分配方案来指定。CIDR 解决膨胀问题。NAT 的主要优势在于它的安装可以不改变路由或主机。本文表述 NAT 的初步设计并讨论它的优势和劣势。

感谢

本文基于 Paul Francis 和 Tony Eng 的论文，出版在《Computer Communication Review》，1993 年 1 月。Paul 引用了 Van Jacobson 的地址重用的概念。

Kjeld Borch Egevang 编辑并生成本文并引入 FTP 序列号的调整。感谢 Jacob Michael Christensen 注释的想法和文本（我们考虑了很久，我们是唯一获得该想法的群体）

1. 简介

当前 IP 因特网面临的两大严重问题是 IP 地址枯竭和路由的膨胀。长期和短期解决方案已经开发出来。短期解决方案是 CIDR（无类域间路由）。长期解决方

案建议采用具有更大地址空间的新的互连协议。

在长期解决方案准备好以前，可能通过地址重用来减少对 IP 地址的需求量。本解决方案的依据是在根域中只有很少比例的主机同时与域外进行通迅（根域是只处理从或到域内的通迅的域，如组织内部网络）。实际上，很多根域内主机从不与域外通迅（如果有也不多）。因此，当需要与外界通迅时，只根域内 IP 地址的一个子集需要转换为全球唯一的 IP 地址。

本方案的缺点是需要去掉 IP 地址的端到端标志，又靠增加网络的状态信息来重新生成它。要减少这个潜在的缺陷需要做大量的工作。实际上，面向连接的协议本质上也是在每次跳跃时进行地址重用。

本方法的巨大优势是它可以不需要修改主机或路由器而进行增量安装。（一些不常用的应用程序可能需要修改）。因此，本方案可快速地实现和实验。如果没有其它更好的办法，在其它的更复杂和长期解决方案生效前，本方案可以工作以提供临时的帮助。

2. NAT 预览

本文中描述的设计称为 NAT，即网络地址转换器。NAT 是一个路由功能，可以如图 1 中一样配置。只有根边界路由需要修改。

NAT 的基本操作如下所述。根域中的地址可以被其它根域重用。例如，单个 A 类地址可以被很多根域使用。在根域到主干网络的每个出口点都配置 NAT。如果出口点超过 1 个，每个 NAT 使用同样的转换表是非常重要的。

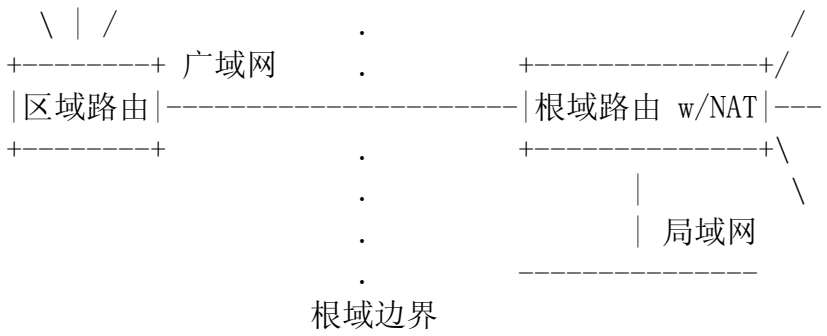

图 1：NAT 配置

例如，在图 2 的例子中，根域 A 和 B 内部都使用 A 类地址 10.0.0.0。根域 A 的 NAT 分配 C 类地址 198.76.29.0，根域 B 的 NAT 分配 C 类地址 198.76.28.0。这些 C 类地址是全球唯一的，其它的 NAT 盒不能使用它们。

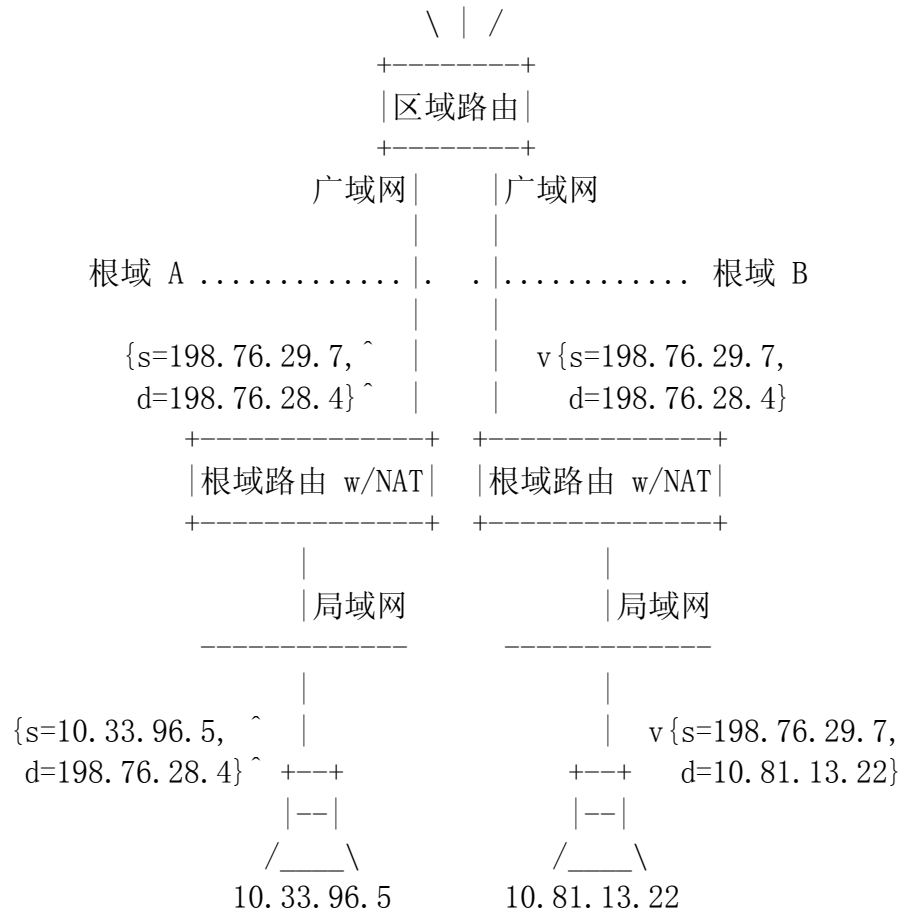

图 2：NAT 的基本操作

当根域 A 内主机 10.33.96.5 希望发送包到根域 B 内主机 10.81.13.22，它使用全球唯一的地址 198.76.28.4 作为目的地址将包发送到它的主路由。根路由器有网络 198.76.0.0 的静态路由，所以包被传到广域网链路中。但是，在包传送前，NAT 将 IP 头部中的源地址 10.33.96.5 转换为全球唯一的 198.76.29.7。同样，IP 包在返加的路径上也要经历同样的地址转换。

要注意，这并不需要修改主机或路由器。例如，如同根域 A 内主机所关心的，198.76.28.4 是在根域 B 内的主机使用的地址。地址转换是完全透明的。

当然，这只是个简单的例子。还有大量的问题需要探索。在下一节中，将讨论 NAT 的各方面的问题。

3. NAT 各方面的问题

3.1 地址空间

可重用和不可重用地址的区分

要使用 NAT 操作正确，有必要区分 IP 地址为两部份：根域内使用的可重用

地址和全球唯一的地址。可重用地址称为本地地址，全球唯一地址程序全球地址。任何一个地址要么是本地地址，要么是全局地址。没有重叠。

重叠的问题如下。要是根域 A 中的主机希望发送包给根域 B 中的一个主机，但是根域 B 中的本地地址与根域 A 中的本地地址是重叠的。在这种情况下，根域 A 的路由于法用它的本地地址区分出根域 B 的全局地址。

初始分配本地和全局地址

局域网应该只分配单个 A 类地址。（见 RFC 1597[3]）该地址可以用来给不需要连接到互联网的网络使用。这样通过将试验地址转换为全球唯一的互联网地址，NAT 就可以很轻松地将将一个实验网络变为“真正的”网络。

现存的分配有唯一地址但快用完的根域可以修改地址子网。余下的空闲地址可用于通过 NAT 与外界的通迅。

3.2 跨 NAT 路由

运行 NAT 的路由器不应该将本地网络通告给主干网络。只有拥有全局地址的网络可以被根域外所知。但是，NAT 从根域边界路由器接收到的全局信息可以在域内正常地通告。

跨越主干网络的私有网络

在很多情况下，私有网络（如组织网络）会分布在不同地方，而且会使用公共主干网络在这些地方之间通迅。在这种情况下，就不值得进行地址转换，既因为有大量的主机希望通过主干网络通迅，这样就需要巨大的地址表，而且因为有大量的应用程序需要配置地址，如反向的域名服务器。称这种私有网络为主干分隔根域。

主干分隔根域可以表现为如同没有被分隔一样。这样所有分区中的路由器应该维护所有分区的本地地址空间路由表。当然，公共主干网络不需要维护任何本地地址路由表。因此，边界路由必须封装通过主干网络的数据，如同打隧道一样。要这样做，每个 NAT 盒需要设置全局地址在隧道一头。当在根域分区 X 中的 NAT 盒 x 需要传递包到根域分区 Y 时，它会将包用目的地址设置为可进行封装的 NAT 盒 y 的全局地址的 IP 头部来封装该包。当 NAT 盒 y 收到有该目的地址的包时，它打开 IP 地址封装，然后在内部路由它。

3.3 头部操控

要修改 IP 地址，NAT 必须悠 IP 校验和和 TCP 校验和。记注，TCP 校验和还覆盖了包括源地址和目的地址的一个假头部。NAT 还必须区分 ICMP 和 FTP 来修改 IP 地址出现的地方。还有其它不确定的地址需要修改。希望大多如这类应用程序可以在 NAT 试验中被发现。

对 IP 和 TCP 的校验和的修改是简单且有效的。既然都使用 1 补码求和，计算转换前和转换后地址的算术差并将其加到校验和上是可行的。唯一需要注意的部份是决定求和的校验和是否有进位（正负两个方向）。如果有，必须加 1 或减 1 以保证 1 补码算法的正确性。下面是示例代码（C 语言）：

```
void checksumadjust(unsigned char *chksum, unsigned char *optr,
int olen, unsigned char *nptr, int nlen)
/* assuming: unsigned char is 8 bits, long is 32 bits.
  - chksum points to the chksum in the packet
  - optr points to the old data in the packet
  - nptr points to the new data in the packet
*/
{
  long x, old, new;
  x=chksum[0]*256+chksum[1];
  x=~x;
  while (olen) {
    if (olen==1) {
      old=optr[0]*256+optr[1];
      x-=old & 0xff00;
      if (x<=0) { x--; x&=0xffff; }
      break;
    }
    else {
      old=optr[0]*256+optr[1]; optr+=2;
      x-=old & 0xffff;
      if (x<=0) { x--; x&=0xffff; }
      olen-=2;
    }
  }
  while (nlen) {
    if (nlen==1) {
      new=nptr[0]*256+nptr[1];
      x+=new & 0xff00;
      if (x & 0x10000) { x++; x&=0xffff; }
      break;
    }
    else {
      new=nptr[0]*256+nptr[1]; nptr+=2;
      x+=new & 0xffff;
      if (x & 0x10000) { x++; x&=0xffff; }
      nlen-=2;
    }
  }
```

```
    x=~x;
    chksum[0]=x/256; chksum[1]=x & 0xff;
}
```

文件传输协议（FTP）中，PORT 命令的参数包括 IP 地址（ASCII 码!）。如果 PORT 命令中的 IP 地址是根域的本地地址，NAT 必须替换它。由于地址是用 ASCII 码编码的，因此替换可能会使用包的长度发生变化（例如，10.18.177.42 有 12 个 ASCII 字符，198.45.228.137 有 14 个 ASCII 字符）。如果新的长度与旧的长度相同，只需要调整 TCP 的校验和。如果新的长度小于旧的长度，可以增长 0 字符，但这并不保证能够工作。如果新的长度大于旧的长度，TCP 的序列号也必须修改。

常使用一个特殊表格来修改 TCP 序列或响应的源端口 FTP 号或目的端口 FTP 号。表格的表项应包括源、目的、源端口、目的端口、初始序列号、动态序列号和一个时间戳。只有当使用 FTP 的 PORT 命令时才会增加新的项。初始序列号用于确认包的序列号是在最后的 FTP 的 PORT 命令之前或之后（变化序列号在每个 FTP 的 PORT 命令之后需有增加）。序列号增加，通知号减少。如果包的 FIN 位设置，关联的项很快就会删除（1 分钟的安全时间）。长时间（如 24 小时）没有使用的项也应该安全地删除。

序列号调整应该仔细编码，一般一应影响 TCP 的性能。当然，如果 FTP 会话是加密的，PORT 命令会失败。

如果要通过 NAT 传输 ICMP 消息，需要修改 2 个地址和 3 个校验和。这是因为绝大多数 ICMP 消息体包括原始 IP 包的部分。因此，NAT 要对主机完全透明，ICMP 包数据部分中嵌入的 IP 头部中的 IP 地址必须修改，该 IP 头部中的校验和必须与修改相一致，且 ICMP 头部的校验和必须反映其消息体中的 IP 头部和校验和的修改。更进一步，如前所述，标准 IP 头部也必须修改。

还不确定是否需要修改 ICMP 消息体中的 IP 头部信息。这取决于是否有主机的代码使用该 IP 头部中的信息。然而，提供确切的头部给路由器或主机对于帮助调试可能会有用处。在任何情况下，Echo 和 Timestamp 消息都不需要修改，且 NAT 也不需要处理 Redirect 消息。SNMP 消息可能要修改，但是否需要修改与 ICMP 消息相比更不确定。

以 IP 地址作内容的应用程序

任何携带 IP 地址的应用程序都不能通过 NAT，除非 NAT 知道该程序且进行适当的转换。要使用 NAT 知道所有这样的程序是不可能的，且是不必要的。而且，如果使用了加密算法，NAT 就不可能完成转换。

如果运行这类系统的主机分配了全局地址，可以使其避免使用 NAT。它是否可以工作取决于内域路由算法的能力和内部网拓扑结构。这是因为全局地址必须在内域路由算法中公告。像 RIP 一样低效的路由算法，需要主机有它自己的 C

类地址空间,它不仅要在内部公告,还需要在外部公告(这样会加重全局的膨胀)。像 OSPF 一样高效的路由算法，主机地址可以个别地传递，且可以来自 NAT 的表格。

隐私、安全和调试考虑

非常不幸,NAT 减少了提供安全特性的选择余地。使用 NAT,通过 IP 地址(如 TCP 头部的校验和）传递的携带有 IP 地址的信息不能加密。绝大多数应用程序的加密工作正常，但它会阻止 TCP 头部的加密。

换局话说，NAT 它自身就提供了某种隐私机制。这是由于主干网中的设备不知道通迅的发送和接收方主机（当然，应假设应用数据是加密的)。

这种特性也使得在增强潜在的隐私同时也会增加调试问题(包括安全侵入检查）的难度。如果主机滥用因特网发生（如尝试攻击其它设备或更有甚者，发送大量的垃圾邮件或其它的东西)，定位问提的根源就更困难了。因为主机的 IP 地址是隐藏的。

4. 结论

NAT 可能是地址枯竭和放大问题的较好的短期解决方案。这是因为它需要非常少的修改，且可以增量式安装。NAT 有一些负面的特性使用其不适合作为长期解决方案，甚至不适作为短期解决方案。只有实现和试验能够决定它的适用性。

负面特性

1. 它需一个稀疏端到端流通矩阵。而且 NAT 表很大，因此它的性能很低。当然预测的该端到端流通矩阵稀疏与否还需要使用 NAT 试验来决定。在其它情况下，将来应用可能需要一个全流通矩阵（如分布式资源搜索)，这样使用得长期使用 NAT 没有吸引力。

2. 它增加丢失地址的概率。

3. 它阻碍特定的程序（或至少使用它们运行困难)。

4. 它隐藏主机的标识。虽然这有助于保护隐，但一般而言负面作用更大。

5. 使用 SNMP，DNS，……，和其它待发现的协议有问题。

当前实现

Paul 和 Tony 在 KA9Q TCP/IP 软件[1]的公共域实现了一个试验性的 NAT 协议。该实现操控地址和 IP 校验和。

Kjeld 在 Cray 通迅 IP 路由器上实现了 NAT。该实现通过了 Telnet 和 FTP 测试。该实现操控地址，IP 校验和，TCP 序列/响应号和 FTP 的 PORT 命令。

这些原型证明解决本文中所述的限制，IP 地址就可被透明地转换。

参考资料

[1] Karn， P.，“KA9Q”，ucsd.edu(hamradio/packet/ka9q/docs)的匿名 FTP。

[2] Fuller，V.，Li，T.和 J. Yu，“无类域间路由（CIDR）和地址分配和集合策略”，RFC 1519，BARRNet，cisco，Merit，OARnet，1993 年 9 月。

[3] Rekhter，Y.，Moskowitz，B.，Karrenberg，D.和 G. de Groot，“私有互联网的地址分配”，RFC 1597，T.J. Watson 研究中心，IBM 公司，Chrysler 公司，RIPE NCC,1994 年 5 月。

安全考虑

本文中没有讨论安全问题。

作者联系方式

Kjeld Borch Egevang
Cray 通迅
Smedeholm 12-14
DK-2730 Herlev
丹麦

电话: +45 44 53 01 00
电邮: kbe@craycom.dk

Paul Francis
NTT 软件试验室
3-9-11 Midori-cho Musashino-shi
东京 180, 日本

电话: +81-422-59-3843
传真: +81-422-59-3765
电邮: francis@cactus.ntt.jp