Network Virtualization

- **Recommended reading**: This note is based on Chapter 19 of the text book.

- **Network virtualization**

  Network virtualization is an abstraction to provide a higher level service which is not available at lower level while hiding the technology details at the lower level. Internetworking is an example of virtualization: routers connect multiple physical networks into a single virtual global network; IP hides the physical network technologies and provides a universal connectivity over the global network. TCP is another example: it provides connection oriented and reliable data delivery service based on a unreliable and connectless data delivery service of IP. We discuss further virtualization from the current Internet based on three technologies: (1) Virtual Private Networks (VPNs), (2) Network Address Translations (NATs), and (3) Overlay Networks.

- **Virtual private networks (VPNs)**

  Many organizations require a private network to guarantee the security in communication. A private network using dedicated physical networks and routers provides a high level of security but the cost is high. A virtual private network (VPN) is a network on the global Internet for secure communication. There are two basic technologies for realizing VPNs: (1) Encryption and (2) Tunnel Technology.

- **VPN tunneling and IP-in-IP encapsulation**

  Assume that an organization has two sites $AS_1$ and $AS_2$ connected by the global Internet. A simple way for a secure communication between the two sites is to encrypt the data in an IP datagram. This simple way, however, does not put the source and destination information in confidential. If a higher level security requires to hide the source and destination information when the datagrams are transmitted over the global Internet, the tunnel technology can be used.

  Assume that sites $AS_1$ and $AS_2$ have the gateways $G_1$ and $G_2$, respectively, to connect the sites to the global Internet. When a host $A$ in $AS_1$ sends an IP datagram $IP_A$ to a host $B$ in $AS_2$, $IP_A$ is first forwarded to $G_1$. $G_1$ encrypts the entire $IP_A$ to get $E(IP_A)$ and generates a new IP datagram $IP_{G_1}$ which has $E(IP_A)$ as data, IP address of $G_1$ as the source address and IP address of $G_2$ as the destination address. $G_1$ forwards $IP_{G_1}$ to $G_1$. On receiving $IP_{G_1}$, $G_2$ decrypts the data of $IP_{G_1}$ ($E(IP_A)$) to get $IP_A$ and forwards it to $B$. This is VPN tunneling and IP-in-IP encapsulation.

- **VPN addressing and forwarding**

  A VPN tunnel between a pair of gateways can be considered as a point-to-point link between the gateways. In the routing table at gateway $G_1$, the next hop router to the networks at site represented by the other gateway $G_2$ is tunnel to $G_2$. See pages 402-403 of the text book for an example of VPN addressing and forwarding. By the VNP tunnel technology, private IP addresses can be used within a site.

- **Network address translation (NAT)**

  NAT is a technology which changes (translates) the IP addresses in outgoing/incoming datagrams at the gateway. Typically, a site using private IP addresses is connected to the global Internet (where the unique global IP addresses are used) by a gateway known as an NAT Box. The NAT
Box has a global IP address. Datagrams from a host to the global Internet have the private IP address of the host as the source IP address. The NAT Box replaces the private source IP address with its global IP address in the datagrams (from site to Internet) and forwards the modified datagrams to the global Internet. Datagrams from the global Internet to a host in the site have the global IP address of the NAT Box as the destination IP address. The NAT Box replaces the global destination address with the private IP address of the destination host in the datagrams (from Internet to the site) and forwards the modified datagrams to the destination host. From the global Internet, the NAT Box is viewed as a single host with a global IP address. From the site, the NAT Box is viewed as gateway to the Internet. NAT provides transparent IP-level access to the global Internet from a host with a private IP address. From the application point of view, the host can use private IP address to communicate with global Internet while from the strict technique point of view, the private IP address is not routable in the global Internet. NAT eliminates this hurdle by the translation between the private address and global address.

- **NAT translation table**

  NAT uses a table to keep information for address translation. There is an entry in the table for each communication session between a host in a site and the global Internet. Main items in the entry for a communication session include:

  - Internal IP, the private IP address used by the internal host for the session.
  - Internal Port, the protocol port number used by the internal host for the session.
  - External IP, the IP address of the external host in the Internet for the session.
  - External Port, the protocol port number used by the external host for the session.
  - NAT Port, the protocol port number used by the NAT Box. This port number is used for communications between the NAT and the external hosts. NAT also use it to distinguish the sessions with a same internal port number.
  - Payload Type, the transport protocol type (e.g., TCP, UDP, ICMP, etc.).

  An example from the text book:

<table>
<thead>
<tr>
<th>Internal IP</th>
<th>Internal Port</th>
<th>External IP</th>
<th>External Port</th>
<th>NAT Port</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.5</td>
<td>38023</td>
<td>128.10.19.20</td>
<td>80</td>
<td>41003</td>
<td>TCP</td>
</tr>
<tr>
<td>192.168.0.1</td>
<td>41007</td>
<td>128.10.19.20</td>
<td>80</td>
<td>41010</td>
<td>TCP</td>
</tr>
<tr>
<td>192.168.0.3</td>
<td>38023</td>
<td>128.210.1.5</td>
<td>80</td>
<td>41007</td>
<td>TCP</td>
</tr>
</tbody>
</table>

  The TCP connection between the internal host 192.168.0.5 and the external host 128.10.19.20 is split into two sessions, one session between the internal host and the NAT identified by (192.168.0.5,38023,128.10.19.20,80), and one session between the NAT and the external host identified by (G,41003,128.10.19.20,80), where G is the global IP address for the NAT.

- **NAT translation table initialization**

  There are three ways to create an NAT translation table: (1) Manual Initialization, network administrators create the table; (2) Outgoing Datagram Initialization, when an internal host initiates a communication session with an external host, a table entry is created for the session based on the outgoing datagram; and (3) Incoming Name Lookups, when an external host looks
up the domain name of an internal host, (internal) DNS sends the address of the NAT Box as the answer and NAT creates a table entry for the internal host.

Each method for creating the translation table has its advantages and disadvantages. By manual initialization, a communication session can be initialized at either external host or internal host. But the table is manually set and does not respond to changes efficiently. It is convenient to manually set-up the table for some well known and permanent communication sessions. Outgoing initialization only allows an internal host to initiate a communication session but the table can be created automatically and dynamically. It is convenient for dynamic accesses from internal hosts to the Internet. Incoming Name Lookups allows an external host to initiate a communication session but the DNS server needs to be specially configured which may not be trivial. The outgoing initialization is the most commonly used way to create the table.

- Other issues with NAT

When the data in a communication session is from ICMP, the NAT Box needs to check the data to decide if the ICMP message should be forwarded. For example, an ICMP echo and echo reply messages should be forwarded, while a re-direct message needs to be checked to decide the forwarding.

For applications using TCP and UDP, NAT Box usually does not look at the application data. However, some applications may carry the IP address and port number information in the application data, for example, the File Transfer Protocol (FTP) first creates a control connection to get parameters (end-points) for creating a new TCP connection for the actual file transfer. NAT Box usually does not support such applications. Many NAT Boxes are specially configured to support some well known applications such as FTP that carry IP addresses and port numbers in application data.

It is difficult for NAT Box to handle fragmented IP datagrams. As we discussed before, NAT uses IP addresses and port numbers for address translation. That means NAT needs to check the TCP/UDP header to get the port numbers. However, only the first piece of a fragmented IP datagram contains the TCP/UDP header and the pieces of the fragmented datagram may arrive out of order. A solution for a fragmented datagram is either the NAT Box performs the reassembly or discards the datagram. Either way is not satisfactory. A common practice is to discard fragmented datagrams.

- Overlay networks

Network virtualization provides abstract services to a subset of hosts. These services are not directly available from the Internet. VPN allows two hosts connected as they were directly connected. NAT allows a set of hosts with private IP addresses to access Internet as they had global IP addresses. Overlay network is a technology which allows several sites connected in the Internet as they were connected by a single physical network.

An overlay network connects multiple sites using the Internet: each site is viewed as a network node, a virtual dedicated link between a pair of nodes is created via VPN. NAT can be used if private addresses are used within a site. A virtual topology is created for an overlay network and the network administrators can change the topology dynamically to provide a best service.