12: VPN, IPV6, NAT, MobileIP

Last Modified:
4/9/2003 1:14:36 PM

Adapted from Gordon Chaffee's slides
http://bmrc.berkeley.edu/people/chaffee/advnet98/

Virtual Private Networks (VPN)

Virtual Private Networks

- Definition
  - A VPN is a private network constructed within the public Internet
- Goals
  - Connect private networks using shared public infrastructure
- Examples
  - Connect two sites of a business
  - Allow people working at home to have full access to company network

How accomplished?

- IP encapsulation and tunneling
- Same as we saw for Multicast
- Router at one end of tunnel places private IP packets into the data field of new IP packets (could be encrypted first for security) which are unicast to the other end of the tunnel

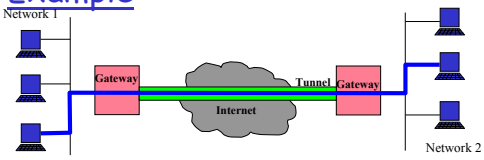
Motivations

- Economic
  - Using shared infrastructure lowers cost of networking
  - Less of a need for leased line connections
- Communications privacy
  - Communications can be encrypted if required
  - Ensure that third parties cannot use virtual network
- Virtualized equipment locations
  - Hosts on same network do not need to be co-located
  - Make one logical network out of separate physical networks
- Support for private network features
  - Multicast, protocols like IPX or Appletalk, etc

Examples

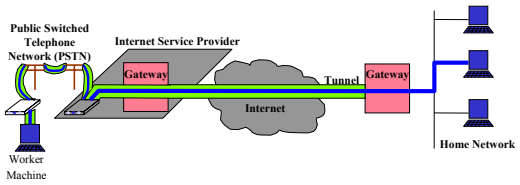
- Logical Network Creation
- Virtual Dial-Up

Logical Network Creation Example



- Remote networks 1 and 2 create a logical network
- Secure communication at lowest level

Virtual Dial-up Example



- Worker dials ISP to get basic IP service
- Worker creates tunnel to Home Network

IPv6

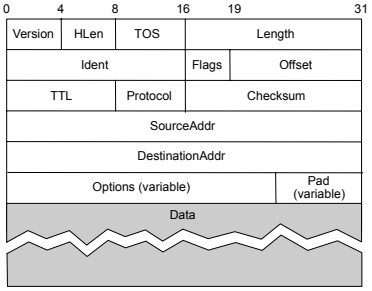
History of IPv6

- IETF began thinking about the problem of running out of IP addresses in 1991
- Requires changing IP packet format - HUGE deal!
- While we're at it, lets change X too
- "NGTrans" (IPv6 Transition) Working Group of IETF - June 1996

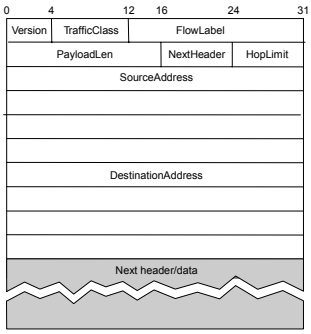
IPv6 Wish List

- From "The Case for IPv6"
- Scalable Addressing and Routing
- Support for Real Time Services
- Support of Autoconfiguration (get your own IP address and domain name to minimize administration
- Security Support
- Enhanced support for routing to mobile hosts

IPv4 Datagram



IPv6 Datagram



IPv6 Base Header Format

- VERS = IPv6
- TRAFFICE CLASS: specifies the routing priority or QoS requests
- FLOW LABEL: to be used by applications requesting performance guarantees
- PAYLOAD LENGTH: like IPv4's datagram length, but doesn't include the header length like IPv4
- NEXT HEADER: indicates the type of the next object in the datagram either type of extension header or type of data
- HOP LIMIT: like IPv4's TimeToLive field but named correctly
- NO CHECKSUM (processing efficiency)

Address Space

- 32 bits versus 128 bits - implications?
  - 4 billiion vesus 3.4 $X10^{38}$
  - 1500 addresses per square foot of the earth surface

Addresses

- Still divide address into prefix that designates network and suffix that designates host
- But no set classes, boundary between suffix and prefix can fall anywhere (CIDR only)
- Prefix length associated with each address

Addresses Types

- Unicast: delivered to a single computer
- Multicast: delivered to each of a set of computers (can be anywhere)
  - Conferencing, subscribing to a broadcast
- Anycast: delivered to one of a set of computers that share a common prefix
  - Deliver to one of a set of machines providing a common servicer

Address Notation

- Dotted sixteen?
  - 105.67.45.56.23.6.133.211.45.8.0.7.56.45.3.189.56
- Colon hexadecimal notation (8 groups)
  - 69DC:8768:9A56:FFFF:0:5634:343
- Or even better with zero compression (replace run of all 0s with double ::)
- Makes host names look even more attractive huh?

Special addresses

- Ipv4 addresses all reserved for compatibility
  - 96 zeros + IPv4 address = valid IPv6 address
- Local Use Addresses
  - Special prefix which means "this needn't be globally unique"
  - Allow just to be used locally
  - Aids in autoconfiguration

Datagram Format

- Base Header + 0 to N Extension Headers + Data Area

Extensible Headers

- Why?
- Saves Space and Processing Time
  - Only have to allocate space for and spend time processing headers implementing features you need
- Extensibility
  - When add new feature just add an extension header type - no change to existing headers
  - For experimental features, only sender and receiver need to understand new header

Flow Label

- Virtual circuit like behaviour over a datagram network
- A sender can request the underlying network to establish a path with certain requirements
  - Traffic class specifies the general requirements (ex. Delay < 100 msec.)
- If the path can be established, the network returns an identifier that the sender places along with the traffic class in the flow label
- Routers use this identifier to route the datagram along the prearranged path

ICMPv6

- New version of ICMP
- Additional message types, like "Packet Too Big"
- Multicast group management functions

Summary like IPv6

- Connectionless (each datagram contains destination address and is routed seperately)
- Best Effort (possibility for virtual circuit behaviour)
- Maximum hops field so can avoid datagrams circulating indefinitely

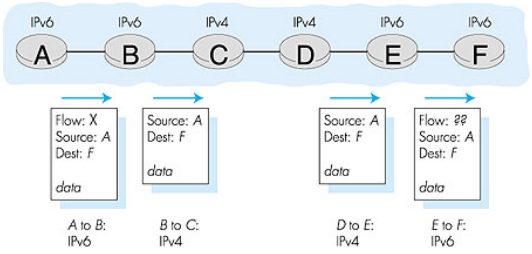
Summary New Features

- Bigger Address Space (128 bits/address)
  - CIDR only
  - Any cast addresses
- New Header Format to help speed processing and forwarding
  - *Checksum*: removed entirely to reduce processing time at each hop
  - No fragmentation
- Simple Base Header + Extension Headers
  - *Options:* allowed, but outside of header, indicated by "Next Header" field
- Ability to influence the path a datagram will take through the network (Quality of service)

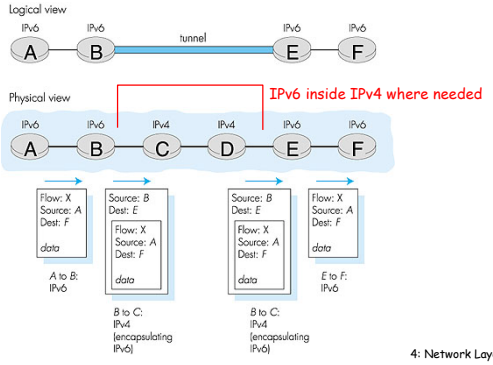
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - no "flag days"
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Two proposed approaches:
  - *Dual Stack*: some routers with dual stack (v6, v4) can "translate" between formats
  - *Tunneling:* IPv6 carried as payload n IPv4 datagram among IPv4 routers

Dual Stack Approach



Tunneling



6Bone

- The 6Bone: an IPv6 testbed
- Started as a virtual network using IPv6 over IPv4 tunneling/encapsulation
- Slowly migrated to native links fo IPv6 transport
- RFC 2471

Recent History

- First blocks of IPv6 addresses delegated to regional registries - July 1999
- 10 websites in the .com domain that can be reached via an IPv6 enhanced client via an IPv6 TCP connection (http://www.ipv6.org/v6-www.html) - it was 5 a year ago (not a good sign?)

IPv5?

- New version of IP temporarily named "IP - The Next Generation" or IPng
- Many competing proposals; name Ipng became ambiguous
- Once specific protocol designed needed a name to distinguish it from other proposals
- IPv5 has been assigned to an experimental protocol ST

Network Address Translation (NAT)

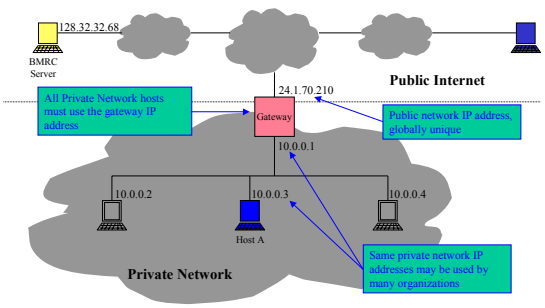
Background

- IP defines private intranet address ranges
  - 10.0.0.0 - 10.255.255.255 (Class A)
  - 172.16.0.0 - 172.31.255.255 (Class B)
  - 192.168.0.0 - 192.168.255.255 (Class C)
- Addresses reused by many organizations
- Addresses cannot be used for communication on Internet

Problem Discussion

- Hosts on private IP networks need to access public Internet
- All traffic travels through a gateway to/from public Internet
- Traffic needs to use IP address of gateway
- Conserves IPv4 address space
  - Private IP addresses mapped into fewer public IP addresses
  - Will this beat Ipv6?

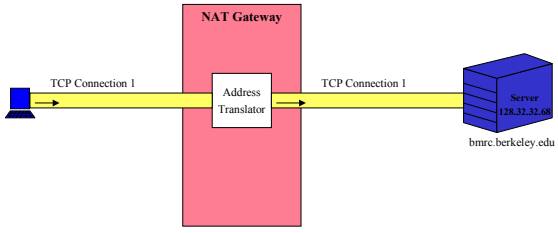
Scenario



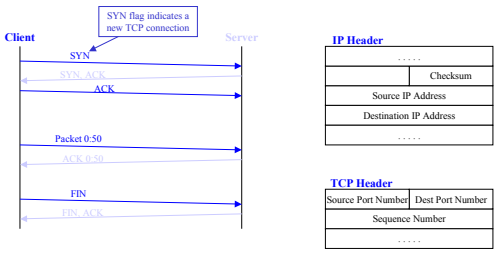
Network Address Translation Solution

- Special function on gateway
  - IP source and destination addresses are translated
  - Internal hosts need no changes
- No changes required to applications
- TCP based protocols work well
- Non-TCP based protocols more difficult
- Provides some security
  - Hosts behind gateway difficult to reach
  - Possibly vulnerable to IP level attacks

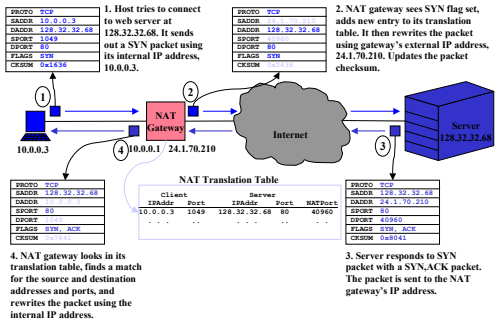
NAT Example



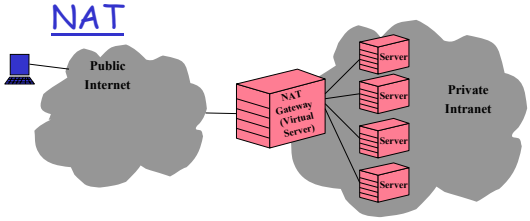
TCP Protocol Diagram



TCP NAT Example

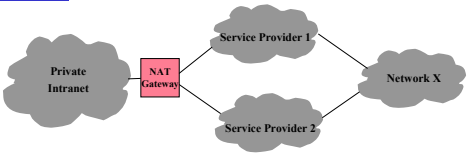


Load Balancing Servers with NAT



- Single IP address for web server
- Redirects workload to multiple internal servers

Load Balancing Networks with NAT



- Connections from Private Intranet split across Service Providers 1 and 2
- Load balances at connection level
  - Load balancing at IP level can cause low TCP throughput

NAT Discussion

- NAT works best with TCP connections
- NAT breaks End-to-End Principle by modifying packets
- Problems
  - Connectionless UDP (Real Audio)
  - ICMP (Ping)
  - Multicast
  - Applications use IP addresses within data stream (FTP)
- Need to watch/modify data packets

MobileIP

MobileIP

- Goal: Allow machines to roam around and maintain IP connectivity
- Problem: IP addresses => location
  - This is important for efficient routing
- Solutions?
  - DHCP?
    - ok for relocation but not for ongoing connections
  - Dynamic DNS (mobile nodes update name to IP address mapping as they move around)?
    - ok for relocation but not for ongoing connections

Mobile IP

- Allows computer to roam and be reachable
- Basic architecture
  - Home agent (HA) on home network
  - Foreign agent (FA) at remote network location
  - Home and foreign agents tunnel traffic
  - Non-optimal data flow

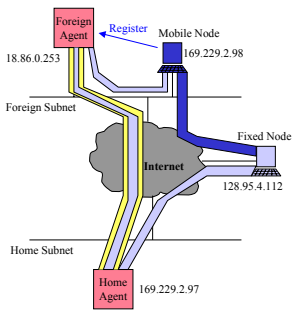
MobileIP

- Mobile nodes have a permanent home address and a default local router called the "home agent"
- The router nearest a nodes current location is called the "foreign agent"
  - Register with foreign agent when connect to network
  - Located much like the DHCP server

Forwarding Packets

- Home agent impersonates the mobile host by changing the mapping from IP address to hardware address ("proxy ARP")
- Sends any packets destined for mobile host on to the foreign agent with IP encapsulation
- Foreign agent strips off and does a special translation of the mobile nodes IP address to its current hardware address

Mobile IP Example



1. The Mobile Node registers itself with the Foreign Agent on the Foreign Subnet. The Foreign Agent opens an IP-IP tunnel to the Home Agent. The Home Agent begins listening for packets sent to 169.229.2.98.

2. The Fixed Node initiates a connection to the Mobile Node. It sends packets to the Mobile Node's home IP address, 169.229.2.98. The packets are routed to the Home Subnet.

3. The Home Agent receives them, encapsulates them in IP-IP packets, and it sends them to the Foreign Agent. Encapsulated packets are addressed to 18.86.0.253.

4. The Foreign Agent decapsulates the IP-IP packets, and it sends them out on the Foreign Subnet. These packets will be addressed to 169.229.2.98.

5. The Mobile Node receives the packets, and it sends responses directly to the Fixed Node at 128.95.4.112.

Avoiding the Foreign Agent

- Mobile host can also obtain a new IP address on the remote network and inform the home agent
- The home agent can then resend the packet to the new IP address

Optimizations

- What if two remote hosts are temporarily close together
- If they want to send traffic to each other, why should it have to go all the way to their home agents and back again
- Optimizations exist to allow the sending node to learn and cache the current location of a recipient to avoid this problem

Roadmap

- Finished with the network layer and IP specifics
- Next on to the link layer
- If two hosts are on the same network how do they send data directly to one another